Logan Wingard | wingarlo

Daniel Ross | rossda

Implementation Assignment 1

1 Linear regression

We made the decision to code in python. The Linear regression program is housingData.py located in the linear directory. To run, use the command:

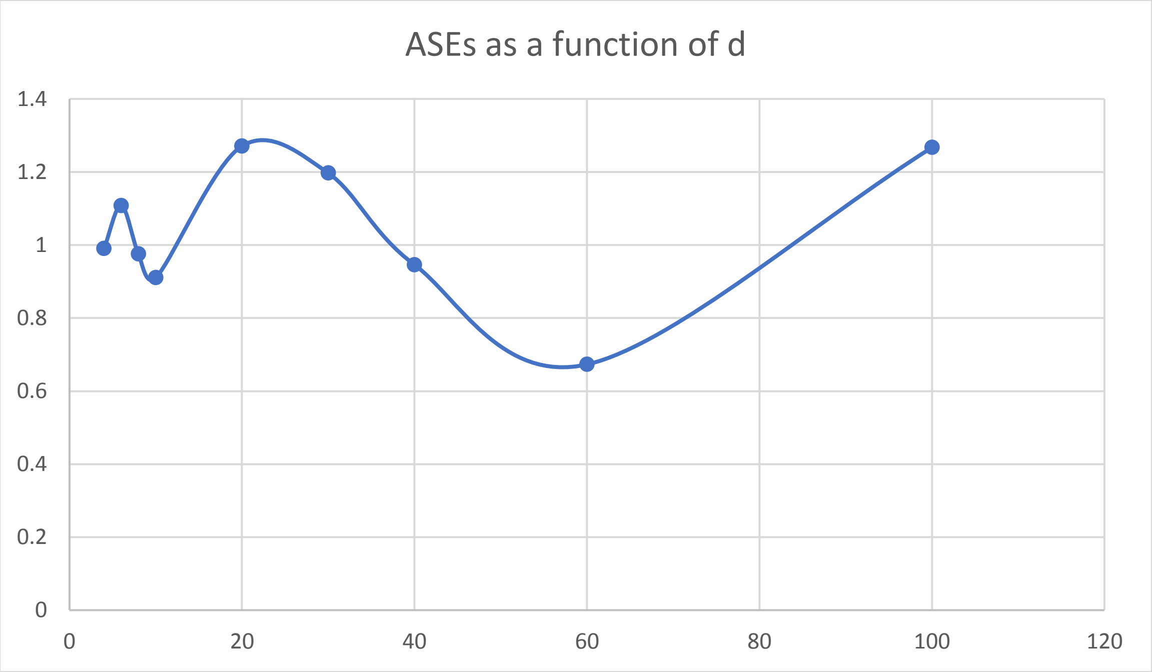
```
python housingData.py
```

The output you will see is the average errors with a dummy variable, without a dummy variable, and then with a dummy variable with added features.

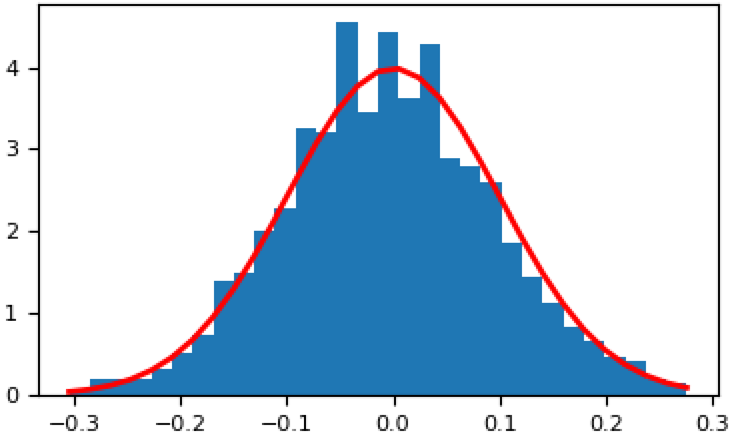
We have three functions for loading data into x and y. We have loadData(filename), loadData2(filename), and adFeature(filename, d). These load data into matrices x and y with a dummy variable, without a dummy variable, and with a dummy variable and added features respectively.

How does removing the dummy variable influence the ASE on the training and testing data?

As expected, the accuracy is higher without the dummy variable. Given that the dummy variable should(ideally) have no influence over the predicted output, any weight multiplier at all would affect the output negatively.



As we can see from the plotted data, the error seems to oscillate around 1 getting further and further away as d gets larger. However, because these are random features sampled from a standard normal distribution, these values change on each run of the program. While the larger number of added features could, by chance, give us a lower error, it could also give us a much higher error. The average of all the tests would be around 1 though.

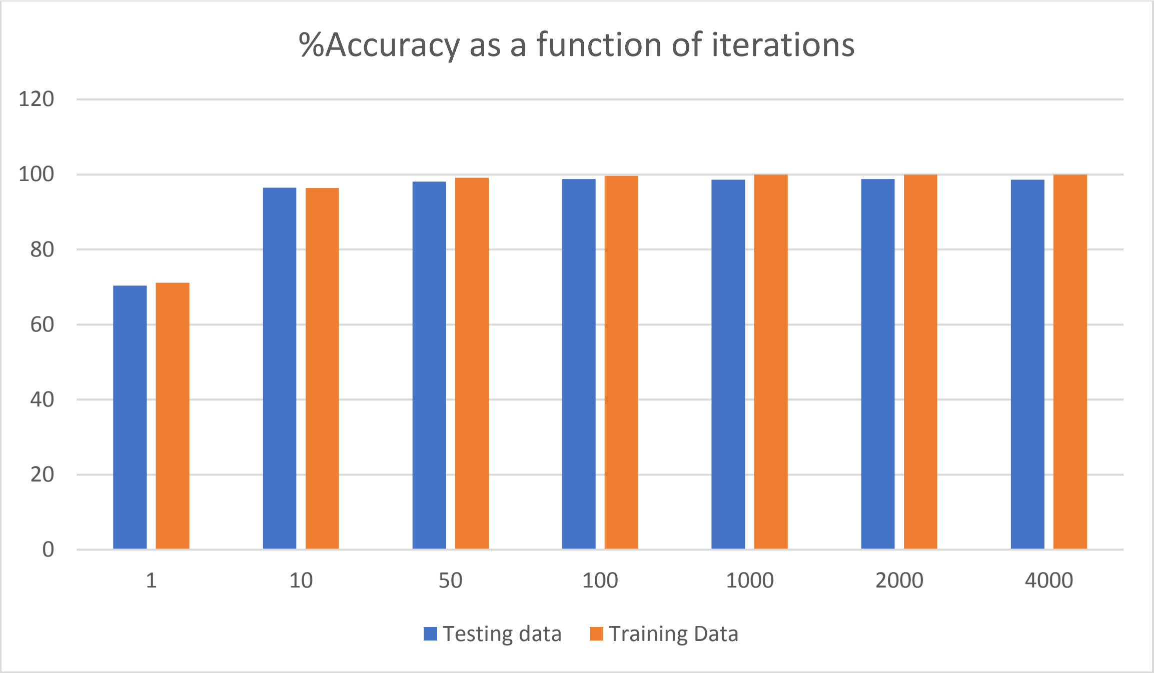


https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html

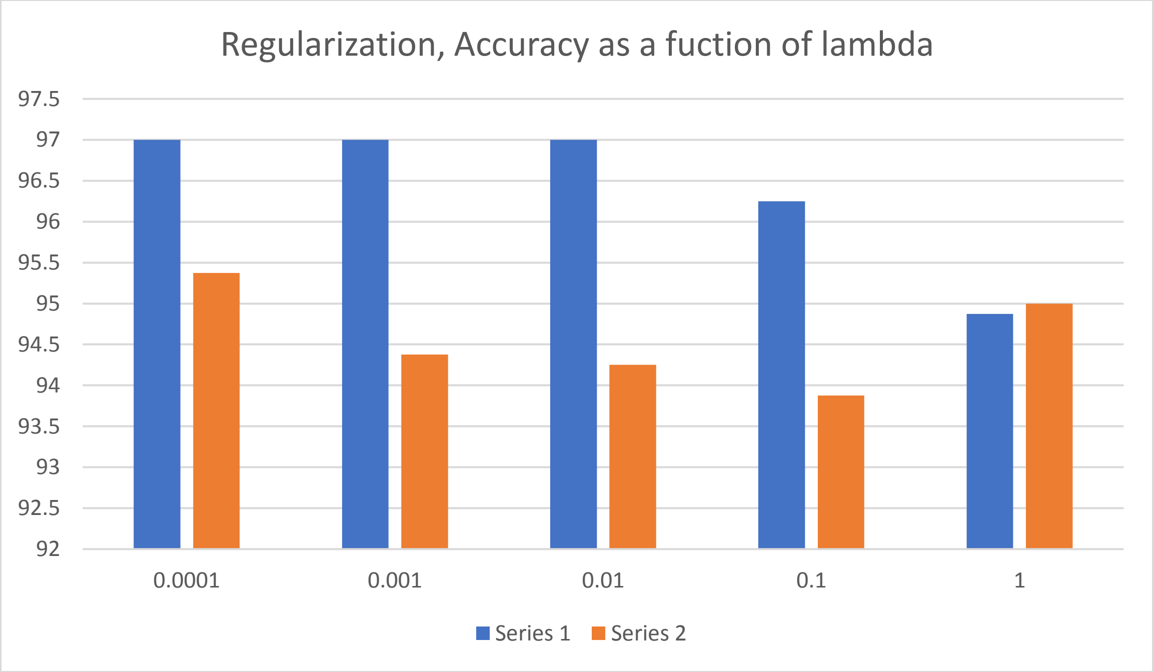
This approximates what our random distribution would look like, taken from the docs.scipy.org website. When we only use 2 samples from this distribution, it is likely that they are very close to 0, meaning they have little to no impact on the predicted output. When we get 100 samples however, there’s a chance of getting a value that has a larger impact on our predicted output, skewing the result.

2 Logistic Regression

For logistic regression, the program is located in the logistic repository in a python file called logreg.py.



As expected, the more iterations the higher the accuracy is. Strangely enough, the % accuracy dipped a tiny bit between 100 and 500 iterations on the testing data for from what I can gather, unknown reasons. I believe the slight drop after 1000 iterations is just due to a 4 looking more like a 9 than a 4 in the testing data.



According to our data, it seems as though as lambda increases, the accuracy of the algorithm decreases. The most likely explanation of this is it is over regulating the relationship between x values and predicted output.